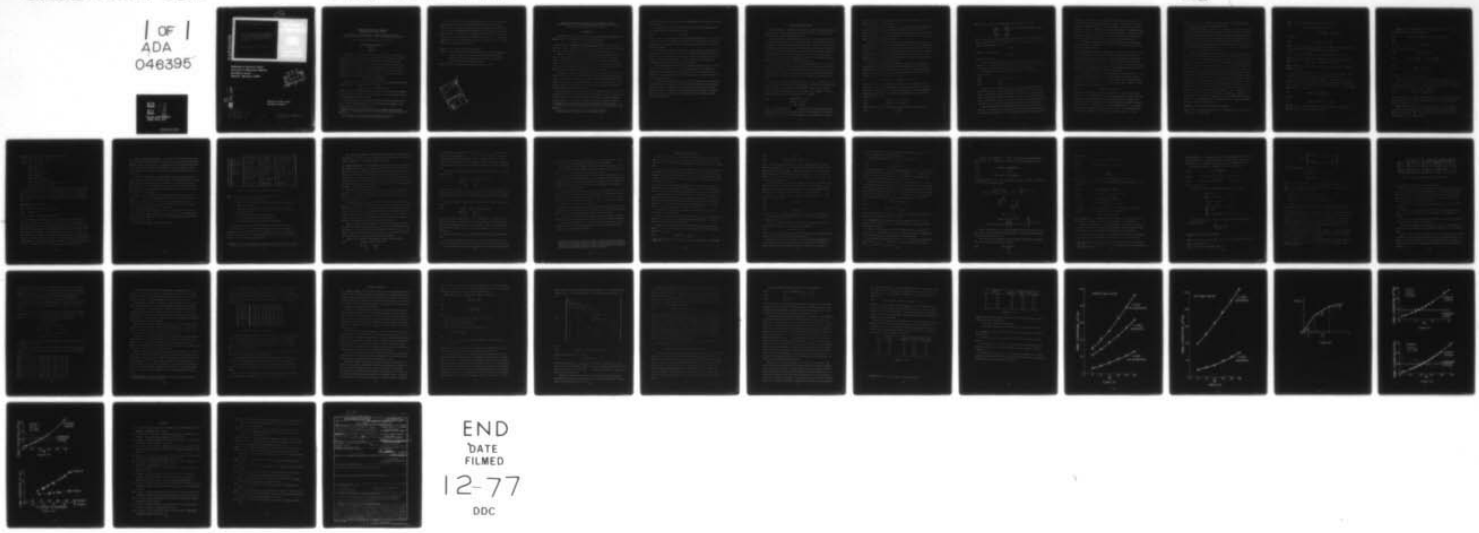
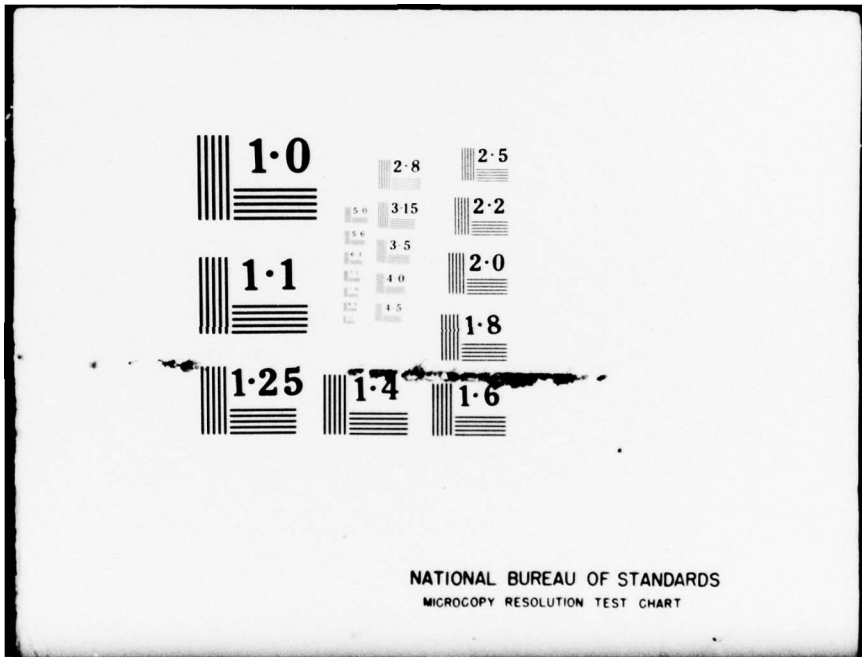
AD-A046 395 WISCONSIN UNIV MADISON MATHEMATICS RESEARCH CENTER F/G 12/1
ON THE SOLUTION OF SOME (PARAMETRIC) LINEAR COMPLEMENTARITY PRO--ETC(U)
AUG 77 J PANG, I KANEKO, W P HALLMAN DAAG29-75-C-0024
UNCLASSIFIED MRC-TSR-1785 NL

1 OF 1
ADA
046395



END
DATE
FILMED
12-77
DDC


1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8
3·15
3·5
4·0
4·5
5·0
5·6
NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

MRC Technical Summary Report #1785

ON THE SOLUTION OF SOME (PARAMETRIC) LINEAR COMPLEMENTARITY PROBLEMS WITH APPLICATIONS TO PORTFOLIO ANALYSIS, STRUCTURAL ENGINEERING AND GRADUATION

Jong-Shi Pang, Ikuyo Kaneko and
Wayne P. Hallman

Mathematics Research Center
University of Wisconsin—Madison
610 Walnut Street
Madison, Wisconsin 53706

August 1977

(Received July 25, 1977)

Approved for public release
Distribution unlimited

Sponsored by

U.S. Army Research Office
P.O. Box 12211
Research Triangle Park
North Carolina 27709

and

National Science Foundation
Washington, D.C. 20550

UNIVERSITY OF WISCONSIN - MADISON
MATHEMATICS RESEARCH CENTER

ON THE SOLUTION OF SOME (PARAMETRIC) LINEAR COMPLEMENTARITY PROBLEMS WITH APPLICATIONS TO PORTFOLIO ANALYSIS, STRUCTURAL ENGINEERING AND GRADUATION

Jong-Shi Pang*, Ikuyo Kaneko** and Wayne P. Hallman**

Technical Summary Report #1785
August 1977

ABSTRACT

In this paper we discuss three important applications of a class of (parametric) linear complementarity problems arising independently from such diverse areas as portfolio analysis, structural engineering and graduation. After explaining how the complementarity problems emerge in these applications, we perform some analytical comparisons (based on operation counts and storage requirements) of several existing algorithms for solving this class of complementarity problems. We shall also present computational results to support the analytical comparisons. Finally, we deduce some conclusions about the general performance of these algorithms.

SIGNIFICANCE AND EXPLANATION

For a given n-vector q and $n \times n$ matrix M, the linear complementarity problem, denoted by (q,M), is to find an n-vector x such that

$$q + Mx \geq 0, \quad x \geq 0 \quad \text{and} \quad x^T(q + Mx) = 0 .$$

The parametric linear complementarity problem consists of the family of linear complementarity problems $(q + \lambda p, M)$ where p is an n-vector and λ is a scalar parameter ranging in a certain subinterval of the real line.

Since its introduction in the sixties, the linear complementarity problem has become an extremely important subject in the field of mathematical programming. It has numerous applications in many different areas.

* Mathematics Research Center, University of Wisconsin, Madison, Wisconsin 53706.
** Department of Industrial Engineering, University of Wisconsin, Madison, Wisconsin 53706.

Sponsored by

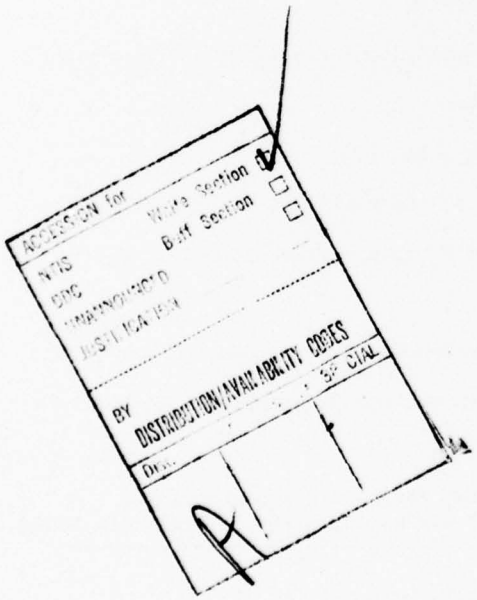
(1) The United States Army under Contract No. DAAG29-75-C-0024 and
(2) The National Science Foundation under Grant No. MCS75-17385.

In this paper, we discuss three important applications of a class of (parametric) linear complementarity problems arising independently from such diverse areas as portfolio analysis, structural engineering and actuarial graduation. Our purposes are: (i) to explain how complementarity problems emerge in these applications, (ii) to compare, analytically and numerically, the computational efficiencies of several existing, "special-purpose" and "general-purpose" algorithms to solve this class of complementarity problems, and (iii) to draw some conclusions on the suitability of these algorithms applied to this particular class of problems.

AMS(MOS) Subject Classifications - 90C20, 90C50, 65K05

Key Words - Applications, (parametric) linear complementarity, quadratic programming, portfolio analysis, index models, structural engineering, graduation, difference-equation method, general-purpose algorithms, special-purpose algorithms, comparison, computational results

Work Unit Number 5 - Mathematical programming and operations research



ON THE SOLUTION OF SOME (PARAMETRIC) LINEAR COMPLEMENTARITY PROBLEMS WITH APPLICATIONS TO PORTFOLIO ANALYSIS, STRUCTURAL ENGINEERING AND GRADUATION

Jong-Shi Pang*, Ikuyo Kaneko** and Wayne P. Hallman**

1. INTRODUCTION

For an n-vector q and n by n matrix M, we denote the linear complementarity problem (LCP) of finding an n-vector x satisfying the conditions

$$q + Mx \geq 0, \quad x \geq 0 \quad \text{and} \quad x^T(q + Mx) = 0$$

by the pair (q,M). The parametric linear complementarity problem (PLCP), denoted by the triple (q,p,M), consists of the family of LCP's $(q + \lambda p, M)$, where p is an n-vector and λ is a scalar parameter ranging in a certain subinterval of the real line.

The purposes of this paper are: (i) to explain some important applications of a class of LCP's and PLCP's characterized by a certain form of the matrix M involved; (ii) to analyze and compare computational efficiencies of several existing, "special-purpose" and "general-purpose" algorithms to solve this class of complementarity problems; (iii) to present computational results and (iv) to draw some conclusions on the suitability of these algorithms applied to this particular class of problems.

In recent years, important applications of the LCP and PLCP have emerged in two entirely different areas. The application of (parametric) quadratic programming to portfolio analysis is well-known. Recently, one of the authors [19] has developed an efficient algorithm to solve a class of simplified but practical portfolio analysis problems. The algorithm is based on a PLCP formulation of the problem and is expected to be considerably more efficient than existing solution methods.

The other application is in the area of structural engineering. Maier (see [12] e.g.) has formulated, as a LCP, the problem of determining the behavior of structures belonging to a certain broad class. Since then, many variations and extensions of this LCP approach have been developed. In particular, one of the authors [10] has proposed a reformulation

* Mathematics Research Center, University of Wisconsin, Madison, Wisconsin 53706.
** Department of Industrial Engineering, University of Wisconsin, Madison, Wisconsin 53706.

Sponsored by

(1) The United States Army under Contract No. DAAG29-75-C-0024 and
(2) The National Science Foundation under Grant No. MCS75-17385.

and algorithm thereof which solves the problem considerably more efficiently than the original method of Maier's.

An important and striking fact is that the LCP's arising in these two applications are defined by matrices of exactly the same form:

$$M = \Sigma + GG^T \tag{1.1}$$

where Σ is an n by n diagonal matrix with positive diagonal entries and G is n by m with m less than n. This implies, in particular, that the special-purpose algorithm developed in [19] for the portfolio analysis application can be used to solve the structural engineering problems.

In this paper, we shall be concerned with the LCP and PLCP which involve a matrix of the form (1.1). Although this class of problems is certainly quite restricted, there seem to be more applications other than the two already mentioned. For instance, we have recently observed that a certain problem in the area of curve fitting and smoothing, or graduation, leads to a LCP with matrix M of the form (1.1).

The form (1.1) implies that the matrix M is symmetric and positive definite. Therefore, the LCP or PLCP with such M can be solved by most well-known LCP algorithms: Notably, Lemke's algorithm [11], Cottle-Dantzig [2] and Graves' [6] principal pivoting algorithm for the (unparameterized) LCP and a parametric version of Graves' algorithm (see [1]) for the parametric LCP.

By taking advantage of the special form (1.1) of M, among other special properties of the problem, the special-purpose algorithms developed for the particular class of complementarity problems are expected to be significantly more efficient than those general-purpose algorithms. The importance of investigating such special methods is justified by the fact this particular class of problems has applications of considerable practical values as mentioned above.

2. PORTFOLIO ANALYSIS APPLICATION

2.1. Problem Formulation. Starting with this section, we present the promised applications of the class of LCP's and PLCP's with matrices of the form (1.1). The first of such applications arises in the area of portfolio analysis.

The standard portfolio analysis problem is one where an investor allocates his initial wealth, taken to be unity with no loss in generality, among n risky securities available in the market. Assuming the security returns are stochastic, Markowitz, in his pioneering work [16] described a theory postulating that rational investors should select a portfolio from the set of all "feasible" portfolios which offers minimum risk (measured by variance) for a given level of expected return, and maximum expected return for a given level of risk. Such a portfolio is said to be efficient. The objective of the portfolio analysis problem is then to determine the set of efficient portfolios.

The above model of Markowitz is known as the mean-variance model. Since its introduction in the fifties, the model has dominated a great deal of the literature in portfolio analysis. As the model is so well-known, we assume that the reader has some basic knowledge about it and therefore shall not discuss its financial and economical implications, among other things. Instead, we proceed directly to the mathematical formulation of the model.

Let a particular portfolio be defined by a vector $x = (x_1,\ldots,x_n)^T$ where x_i represents the proportion of the investor's wealth invested in security i. Let the security returns be represented by a vector of random variables $R = (R_1,\ldots,R_n)^T$ with means $\mu = (\mu_1,\ldots,\mu_n)^T$ and covariance matrix $V = (\sigma_{ij})$ where $\sigma_{ij} = \mathrm{Cov}(R_i,R_j)$. Then the portfolio analysis problem is to find that value of x that will

$$\text{(2.1)} \qquad \begin{aligned} &\text{minimize } \frac{1}{2}x^T Vx - \theta\mu^T x \\ &\text{subject to } \quad a \geq x \geq 0 \\ &\text{and } \quad e^T x = 1 . \end{aligned}$$

Here, e is the vector of one's, a is a given positive vector of upper bounds on the proportions to be invested in the securities, and θ is the coefficient of risk aversion. By varying θ from 0 to ∞, all efficient portfolios can be determined. The upper

bounding constraints exist because of legal, personal or institutional reasons. Some components of the vector a may be infinity.

For every fixed value of θ, problem (2.1) is a (convex) quadratic program which can be solved by a number of algorithms: notably, Markowitz' critical line method [17], Wolfe's algorithm [27], Lemke's algorithm [11] and Von Hohenbalken's algorithm [26]. Both Markowitz' and Wolfe's algorithms are parametric (i.e. will find the solution for every θ), whereas Lemke's and Von Hohenbalken's are not.

Despite the fact that (2.1) is a rather simple (parametric) quadratic program, few (if any) of the currently available quadratic programming algorithms can deal with problems with a large number of securities, an important reason being the fact that typically the covariance matrix V is highly dense, thus causing computer storage problems. In addition to this computational limitation, the general mean-variance model also confronts the informational difficulty of data collection and statistical estimation of the covariances of the security returns in large-scale applications.

To simplify the computational and informational complexity of the general model, Sharpe [24], [25] proposed some simplified models for portfolio analysis. These are the index models which are by far the most widely used formulation of the portfolio analysis problem. In a typical m-index model, it is assumed that the security returns are determined by the values of m market indices. Specifically,

$$R_i = \mu_i + b_{i1}I_1 + \ldots + b_{im}I_m + \varepsilon_i, \quad i = 1,\ldots,n \tag{2.2}$$

where I_j's are the market indices which are random variables with mean zero, b_{ij}'s are constants, and ε_i's are random variables with mean zero and standard deviation σ_i's. These latter random variables ε_i's are used as measures of the errors of approximating the actual security returns R_i's by taking linear combinations of the market indices. The model further assumes

$$\text{Cov}(\varepsilon_i,\varepsilon_j) = 0 \quad \text{for } i \neq j \tag{A.1}$$

$$\text{Cov}(\varepsilon_i,I_j) = 0 \quad \text{for all } i,j. \tag{A.2}$$

Under these two assumptions, the covariance matrix of security returns is given by

$$V = \Sigma + BCB^T \tag{2.3}$$

where Σ is the n by n diagonal matrix whose i-th diagonal entry is σ_i^2,

$$B = \begin{bmatrix} b_{11} & \cdots & b_{1m} \\ \vdots & & \vdots \\ b_{n1} & \cdots & b_{nm} \end{bmatrix}$$ is the $n \times m$ matrix of constants b_{ij}'s ,

and C is the m by m covariance matrix of the market indices. It would be convenient for us to assume that the σ_i's are positive. Since C is symmetric and positive semi-definite, we may decompose C into

$$C = FF^T .$$

Letting $G = BF$, we may write

$$V = \Sigma + GG^T$$

which is of the form (1.1).

In the rest of this subsection, we sketch how problem (2.1) can be solved via the PLCP approach. To simplify the discussion, we assume that all components of the vector a of upper bounds are infinity and refer the reader to [19] for the general case.

The Karush-Kuhn-Tucker optimality conditions for (2.1) are to find a vector x and scalar λ such that

(2.4a) $$u = -\lambda e - \theta\mu + Vx \geq 0, \quad x \geq 0$$

(2.4b) $$u^T x = 0$$

(2.4c) $$e^T x = 1 .$$

To solve (2.4), one would normally treat λ as a variable. However, one may also treat it as a parameter. In this latter situation, (2.4a) and (2.4b) become, for every fixed θ, the defining conditions for the PLCP $(-\theta\mu,-e,V)$ with λ as the parameter. Now if we solve this latter PLCP for all values of λ and if we can determine a suitable value λ^* such that the solution $\bar{x}(\lambda^*)$ to the LCP $(-\lambda^* e - \theta\mu,V)$ also satisfies (2.4c), then $\bar{x}(\lambda^*)$ and λ^* will furnish a solution to (2.4) for this particular value of θ .

The actual application of the PLCP technique to solve (2.4) for all nonnegative values of θ is divided into two stages; the first stage being the determination of the initial solution corresponding to $\theta = 0$ and the second stage being the solution of the

problem for all positive θ. We have seen how the first stage can be carried out. The second stage starts after the initial solution is obtained. The parameter θ is introduced back in the problem, with its value to be increased. The problem now involves two parameters λ and θ. Nevertheless, it is possible to eliminate λ (using (2.4c)) and express it in terms of θ and the values of the basic x-variables. Therefore, the problem is again left with one parameter and the solution process can be completed by treating θ as the only parameter. For a more detailed discussion of the method, see [19].

The PLCP approach described above is applicable to the portfolio analysis problem (2.1) with a general, nonsingular covariance matrix V. In this paper, we shall be concerned only with the index models.

2.2. Compact Inverse Principal Pivoting Algorithm. The purpose of this subsection is to describe a special-purpose algorithm for solving the class of LCP's and PLCP's having matrices of the form (1.1). The algorithm is based on a modification of a parameterized version of Graves' principal pivoting algorithm [6] for solving a PLCP (q,p,M) with a (symmetric and) positive definite matrix M. Since the parametric Graves' algorithm is also the basis for the "condensed Graves' algorithm" to be described in the next section, it is useful for us to review briefly this basic parametric algorithm.

Consider a PLCP (q,p,M) with a general (symmetric and) positive definite M. According to a fundamental theorem in the theory of the LCP [22], there exists a unique solution $\bar{x}(\lambda)$ to $(q + \lambda p, M)$ for every λ. The parametric Graves' algorithm will determine the solution $\bar{x}(\lambda)$ for all λ in a finite number of steps. Suppose for convenience, that a value $\bar{\lambda}$ is obtained such that

$$q + \lambda p \geq 0 \quad \text{for all} \quad \lambda \leq \bar{\lambda} \ . \tag{2.5}$$

This implies, in particular, that $\bar{x}(\lambda) = 0$ for all $\lambda \leq \bar{\lambda}$. Now we want to determine $\bar{x}(\lambda)$ for $\lambda > \bar{\lambda}$. To achieve this, we perform a minimum ratio test to determine how much the value of λ can be increased without violating condition (2.5). Assuming nondegeneracy, we either obtain a unique index where the minimum ratio is attained, or arrive at a nonnegative p. In the first case, we perform a principal pivot on M, obtaining a principal pivotal transform of M and new vectors q and p. This then completes an iteration of the algorithm. In the second case, a further increase of λ can only

increase the values of the current basic variables; therefore the algorithm terminates. For a more detailed description of the algorithm, see [1].

Two remarks are important. Firstly, if one is interested in the solution $\bar{x}(\lambda)$ for λ lying in some bounded interval only, one should incorporate an additional stopping rule so that the algorithm will also terminate when the right end of the interval is reached, in the case where this latter situation occurs before the arrival of a nonnegative p. Secondly, an ordinary LCP (q,M) with a (symmetric and) positive definite M can be solved by the parametric Graves' algorithm as well. To do this, set up the PLCP $(f,q-f,M)$ where f is an arbitrary nonnegative vector such that $f \geq q$. Clearly $\bar{\lambda} = 0$ satisfies (2.5). When λ reaches 1, the problem (q,M) is solved.

Compared to the compact inverse principal pivoting algorithm to be presented shortly, the parametric Graves' algorithm is more general in the sense that it solves any (P)LCP as long as the matrix is (symmetric and) positive (semi-) definite (or has positive principal minors). Being general in nature, it does not take advantage of any possible special structure of the matrix M, except possibly sparsity. In [19], one of the authors has proposed a modified version of the algorithm in such a way that any special structure of M will be preserved and may be used profitably in the entire solution process. Roughly speaking, this is achieved by replacing the pivot steps in the original version by updating the index set of basic variables. Here, we shall not describe the modified algorithm. Instead, we present as Algorithm I below its specialization to a PLCP (q,p,M) with M of the form (1.1). We need to introduce some notations first. Let A be an n by m matrix and $\alpha \subseteq \{1,\ldots,n\}$, $\beta \subseteq \{1,\ldots,m\}$. By $A_{\alpha\beta}$ we denote that submatrix of A consisting of entries whose row and column indices belong to α and β respectively. If $\alpha = \{1,\ldots,n\}$ ($\beta = \{1,\ldots,m\}$), we write $A_{.\beta}$ ($A_{\alpha.}$) to mean $A_{\alpha\beta}$. Similarly, if q is an n-vector, by q_α we mean that subvector of q whose components have indices belonging to α. We attach subscripts to the identity matrix I to denote its dimension. We assume condition (2.5) holds for some $\bar{\lambda}$.

Algorithm I. Compact inverse principal pivoting algorithm.

Step 0. Let $\lambda_{old} = \min_{p_i < 0} (-q_i/p_i)$ and let k be a minimizing index. Let $\alpha = \{1,\ldots,n\} \setminus \{k\}$ and $\beta = \{k\}$. Go to Step 1.

Step 1. Solve the systems of linear equations for o^{q_β} and o^{p_β}

$$(I_m + (G_{\beta.})^T \Sigma_{\beta\beta}^{-1} (G_{\beta.}))\ (o^{q_\beta}, o^{p_\beta}) = \tag{2.6}$$

$$((G_{\beta.})^T \Sigma_{\beta\beta}^{-1} q_\beta,\ (G_{\beta.})^T \Sigma_{\beta\beta}^{-1} p_\beta)$$

and compute

$$s = q - Go^{q_\beta}\ ,\quad t = p - Go^{p_\beta}\ . \tag{2.7}$$

Step 2. If $t_\alpha \geq 0$ and $t_\beta \leq 0$, set $\bar{x}(\lambda) = 0$ and $\bar{x}_\beta(\lambda) = -\Sigma_{\beta\beta}^{-1}(s_\beta + \lambda t_\beta)$ for all $\lambda \geq \lambda_{old}$ and terminate. Otherwise determine the new critical value

$$\lambda_{new} = \min\{\min\{-\frac{s_i}{t_i} : i \in \alpha,\ t_i < 0\},\ \min\{-\frac{s_i}{t_i} :\ i \in \beta,\ t_i > 0\}\}\ , \tag{2.8}$$

and let the new critical index k be a minimizing index. Put $\bar{x}_\alpha(\lambda) = 0$ and $\bar{x}_\beta(\lambda) = -\Sigma_{\beta\beta}^{-1}(s_\beta + \lambda t_\beta)$ for $\lambda \in [\lambda_{old}, \lambda_{new}]$. Set $\lambda_{old} = \lambda_{new}$, go to Step 3.

Step 3. Replace α and β by $\alpha \cup \{k\}$ and $\beta \setminus \{k\}$ ($\alpha \setminus \{k\}$ and $\beta \cup \{k\}$) respectively, depending on $k \in \beta(\alpha)$.

The stopping criterion in Step 2 is set up for solving the PLCP completely. Other stopping rules should of course be included to deal with other situations.

The largest portion of work required in Algorithm I is the execution of Step 1. In fact, the overall efficiency of the algorithm is completely determined by how fast Step 1 can be carried out. In what follows, we demonstrate how this important step can be executed most efficiently. To this end, let

$$A(\beta) = I_m + (G_{\beta.})^T \Sigma_{\beta\beta}^{-1} (G_{\beta.}) \tag{2.9}$$

and write

$$s(\beta) = q - Go^{q_\beta}\ ,\quad t(\beta) = p - Go^{p_\beta}\ . \tag{2.10}$$

The matrix $A(\beta)$ is symmetric, positive definite and has spectral radius at least 1. Moreover, $A(\beta)$ is of order m which is a constant.

Suppose β' and β are two index sets such that $(\beta' \setminus \beta) \cup (\beta \setminus \beta') = \{k\}$. Then by an easy calculation, it can be shown that

$$s(\beta') = s(\beta) - K(\beta,k)\xi^{\beta,k}, \quad t(\beta') = t(\beta) - K'(\beta,k)\xi^{\beta,k} \tag{2.11}$$

where

$$\xi^{\beta,k} = G\eta^{\beta,k} \tag{2.12}$$

with $\eta^{\beta,k}$ the (unique) solution to

$$A(\beta)\,\eta^{\beta,k} = (G_{k.})^T ; \tag{2.13}$$

$$K(\beta,k) = \frac{(s(\beta))_k}{\omega\sigma_k^2 + \xi_k^{\beta,k}}, \qquad K'(\beta,k) = \frac{(t(\beta))_k}{\omega\sigma_k^2 + \xi_k^{\beta,k}} \tag{2.14}$$

with

$$\omega = \begin{cases} 1 & \text{if } k \in \beta' \setminus \beta \\ -1 & \text{if } k \in \beta \setminus \beta' ; \end{cases} \tag{2.15}$$

and finally, σ_k^2 is the k-th diagonal entry of Σ.

Formula (2.11) shows that if $s(\beta)$ and $t(\beta)$ are known, one can compute $s(\beta')$ and $t(\beta')$ by first solving for $\eta^{\beta,k}$ in (2.13), and then computing $\xi^{\beta,k}$, $K(\beta,k)$ and $K'(\beta,k)$ using (2.12), (2.14) and (2.15). A fast approach to solve for $\eta^{\beta,k}$ is to apply Cholesky factorization to the matrix $A(\beta)$. Observing that

$$A(\beta') = A(\beta) + \omega\sigma_k^{-2}(G_{k.})^T (G_{k.}) \tag{2.16}$$

which identifies $A(\beta')$ as a rank-one modification of $A(\beta)$, we may apply a very efficient algorithm developed in [5] to update the Cholesky factors of $A(\beta')$. We rephrase this algorithm in Subroutine I below.

Assume that the Cholesky factors of $A(\beta)$ are known, viz. $A(\beta) = LDL^T$ where L is a unit lower triangular matrix and D is a positive diagonal matrix, we wish to determine the factors $A(\beta') = \bar{L}\bar{D}\bar{L}^T$. In the subroutine below, the matrices $\bar{L}$ and $\bar{D}$ have overwritten L and D respectively.

Subroutine I. (Gill et al. [5]) Updating the Cholesky factors.

1. Define gamma = $\omega\sigma_k^{-2}$ abd $Z = (G_{k.})^T$.

2. For $j = 1,\ldots,m$, compute

```
dsave = d(j)
temp = gamma * Z(j)
d(j) = d(j) + temp * Z(j)
beta = temp/d(j)
gamma = dsave * gamma/d(j)
Z(k) = Z(k) - Z(j) * ℓ(k,j)
ℓ(k,j) = ℓ(k,j) + beta * Z(k)
```

$k = j + 1, \ldots, m$.

Since both $A(\beta)$ and $A(\beta')$ have spectral radii at least 1, the above procedure is numerically stable. Once the Cholesky factors of $A(\beta)$ are known, the solution for $\eta^{\beta,k}$ in (2.13) involves the solution of two unit triangular systems of linear equations and n divisions. More precisely, the computation of $\eta^{\beta,k}$ consists of the following three steps:

Subroutine II. Solving (2.13).

Step 1. Solve for δ in the system $L\delta = (G_{k.})^T$

Step 2. Compute $D^{-1}\delta$.

Step 3. Solve for $\eta^{\beta,k}$ in the system $L^T \eta^{\beta,k} = \delta$.

We now describe how the above analysis is incorporated in the implementation of Algorithm I. Initially, β is the singleton $\{k\}$; $s(\beta)$ and $t(\beta)$ are computed from (2.11) by setting $\beta' = \beta$ and $\beta = \phi$. Note that $s(\phi) = q, t(\phi) = p$ and $\eta^{\phi,k} = (G_{k.})^T$. In general, at the beginning of each iteration, we have $s(\beta)$ and $t(\beta)$ available. The algorithm then goes through the ratio test to determine if termination is reached or a new critical index k is found. In the latter case, a new index set β' is obtained by inserting or deleting the index k to or from β . We then use Subroutine I to update the Cholesky factors of $A(\beta)$, use Subroutine II to solve for $\eta^{\beta,k}$ and finally, compute the new $s(\beta')$ and $t(\beta')$ by (2.11). This completes an iteration of the algorithm.

2.3. Analytical Comparison of Algorithms. In this subsection, we make some comparisons between the compact inverse and some general-purpose LCP algorithms to solve the (parametric) LCP having a matrix of the form (1.1). As "representatives" of general-purpose algorithms, we have chosen Lemke's algorithm for the (nonparametric) LCP and the parametric Graves' algorithm for the PLCP. Clearly, the compact inverse algorithm is applicable to both the LCP and PLCP.

As criteria of the comparison, we shall consider the computer storage requirement (for the input data and "basis inverse representation") and the number of operations (multiplications and divisions) required to solve the problem. Considering the fact that all three algorithms are based on essentially the same principles, i.e. simplex pivoting, results with respect to these two criteria should provide a reasonable measure for the overall performance of the algorithms.

Needless to say, the performance of an algorithm critically depends on how it is implemented. For the purpose of the comparison, no consideration is given on sparsity of problem data. Indeed, in portfolio analysis and the engineering application (§3), the data are highly dense. Also, for the sake of simplicity, we assume that no basis reinversion will be performed in the course of calculations.

A point in which major differences could occur is the representation of basis inverse. In the compact inverse algorithm, it is assumed to be exactly as specified above. In Lemke's and Graves' algorithms, we assume the standard product form using an "eta-file", as described in [4, p. 264].

The comparison results are summarized in Table 2.1.

		compact inverse	Lemke[1]	Graves
storage[2]	input data[3]	q,p,Σ,G: $n(m+3)$	$q+\lambda^* p, M$: n^2+n	q,p,M: n^2+2n
	basis inverse representation[4]	L: $m(m-1)/2$ D: m	eta-file[5]: $T'n$	eta-file[6]: Tn
	total	$n(m+3)+m(m+1)/2$	$n^2+n(T'+1)$	$n^2+n(T+2)$
operation counts	k^{th} iteration (after k-1 pivots)	Subroutine I: m^2+4m Subroutine II: m^2 s and t: $n(m+2)+2$ ratio test: n	ratio-test: n eta-vector: n RHS: n pivot column: kn	ratio-test: n pivot column: $(k-1)n$ eta-vector: n RHS: $2n$
	total	$T[n(m+3)+2(m+1)^2]$	$n\left[\frac{T'(T'-1)}{2}\right]+3T'n$	$n\left[\frac{T(T-1)}{2}\right]+3Tn$

Table 2.1. Storage and operation counts for three algorithms

Notes:
1. Lemke's algorithm is applied to the problem with $\lambda = \lambda^*$.
2. Only real arrays are taken into account.
3. Symmetry of M could be used to reduce the storage requirement in Lemke's and Graves' algorithms.
4. In practice, the length of the eta-file can be controlled by reinversion and/or other devices.
5. T' = total number of pivots in Lemke's algorithm.
6. T = total number of pivots in G aves' algorithm.
7. The operation counts for the ratio-tests are upper bounds.

From the above comparison results, the following two observations can be made:

(i) The efficiencies of the two general-purpose algorithms are about the same provided that the numbers of pivots to solve the problem for each of the two algorithms are close to each other[1] (there is no reason to believe that one algorithm terminates with consistently fewer or more pivots than the other).

1 The number of pivots in Lemke's algorithm depends on the choice of the "covering vector". If the covering vector is chosen to be $q + \bar{\lambda}p$ which is nonnegative then $T = T'$ holds.

(ii) The relative performance of the special- and general-purpose algorithms depend on the values of T (or T') and m/n; the advantage of the special-purpose algorithm becomes more significant as m/n becomes smaller and T larger.

2.4. Computational Results. In this subsection, we report the computational results with the solution of some variations of the portfolio analysis problem (2.1) under the assumptions of an m-index model. Before presenting the results, we point out some further characteristics of the matrix $\Sigma + GG^T$ appearing in this particular application. First of all, the matrix G tends to be highly dense because there is no reason to assume that the indices are uncorrelated. Secondly, the ratio m/n is typically very small. In a potential large-scale application, one might expect $n = 1000$ and $m = 10$. Thirdly and lastly, the number n could be very large in practice.

The experiments performed in the present and later sections were all done on the UNIVAC 1110 at the Madison Academic Computing Center at the University of Wisconsin-Madison. All the input data were generated randomly. The computations in this section were done in double precision.

The main purpose of the experiments below was to test the capacity and efficiency of the compact inverse algorithm in solving practical, large-scale portfolio analysis problems. The fact that the problem data tend to be highly dense makes the application of general-purpose algorithms impractical, if not impossible, for solving large-scale problems since in these algorithms, all nonzero elements of M must be stored. For this and other reasons, we shall not report here any computational results using the general-purpose algorithms. See conclusion (ii) in Section 2.5 and footnote 2 below.

Three sets of experiments were performed, each of which corresponded to solving a variation of the portfolio analysis problem. A number of problems corresponding to different values of n were solved in each set of experiments. For each value of n, the integer m was allowed to vary from five to thirty, in the scale of five, in order to test its influence on the efficiency of the special-purpose algorithm.

The first set of experiments was designed to solve the following problem:

$$\begin{aligned} (2.17) \qquad \text{minimize} \quad & \frac{1}{2} x^T(\Sigma + GG^T)x + q^T x \\ \text{subject to} \quad & x \geq 0 \\ \text{and} \quad & e^T x = 1 \end{aligned}$$

which corresponds to problem (2.1) for a fixed value of θ, with V of the form (1.1) and no explicit upper bounds (i.e. $a_i = \infty$ for all i). The compact inverse algorithm was used to solve the PLCP formulation of (2.17). Two representative sets of results, corresponding to two values of n, are contained in Figure 2.1. We have used time/iteration as a measure of the efficiency of the method. The timing is exclusive of input/output.

The second set of experiments was concerned with solving the problem (2.17) in parametric form, i.e.

$$(2.18) \qquad \begin{aligned} \text{minimize} \quad & \frac{1}{2} x^T(\Sigma + GG^T)x + q^T x + \theta p^T x \\ \text{subject to} \quad & x \geq 0 \\ \text{and} \quad & e^T x = 1 \quad . \end{aligned}$$

The compact inverse algorithm was used to solve the PLCP formulation of the problem. One representative set of results is presented in Figure 2.1. In computing the time/iteration in these experiements, only the second stage of the PLCP approach is taken into consideration.

The third set of experiments was designed to solve problem (2.18) with explicit upper bounds:

$$(2.19) \qquad \begin{aligned} \text{minimize} \quad & \frac{1}{2} x^T(\Sigma + GG^T)x + q^T x + \theta p^T x \\ \text{subject to} \quad & a \geq x \geq 0 \\ \text{and} \quad & e^T x = 1 \quad . \end{aligned}$$

The portfolio analysis problem (2.1) has $q = 0$. Nevertheless, q was not set to be zero in the experiments. An extension of the compact inverse algorithm [19] was used in this instance. The upper bounds a_i were all set equal to $1/20$. Two representative sets of results are shown in Figure 2.2. The timing is computed as in the last set of experiments.

Several important points about the above three sets of experiments are summarized below:

(i) In all these experiments, the values of m were set not to exceed thirty in order to be consistent with the smallness of the ratio m/n in most practical situations.

(ii) In solving (2.18) (and (2.19) as well) by the PLCP approach, the time/iteration

in the first stage is slightly less than the time/iteration in the second stage. This is because in the second stage, somewhat more quantities have to be computed.

(iii) In addition to those problems whose results are plotted in Figures 2.1 and 2.2, we have solved several other problems of the same nature. All of them produce curves similar to those in the two figures. Therefore they are omitted.

(iv) The results of the experiments are consistent with the operation counts of the compact inverse algorithm given in Table 2.1. In particular, the table indicates that the operation counts per iteration of the algorithm is independent of the total number of iterations and quadratic in m. These two facts are substantiated by the two figures (and other results not reported here).

(v) The total numbers of iterations required to solve the problems (2.17)-(2.19) depend very much on the exact type and size of the problems. The largest number of iterations (obtained in the experiments) for (2.17) is 72 when $n = 300$, $m = 30$; that for (2.18) is 131 (excluding the first stage) when $n = 500$, $m = 30$; and that for (2.19) is 201 (excluding the first stage) when $n = 500$, $m = 20$.

2.5. Conclusions. Based on the analysis and computational experience we have gathered, we may draw the following conclusions about the suitability of algorithms to solve practical large-scale portfolio analysis problems of the kinds given by (2.17)-(2.19).

(i) The compact inverse algorithm is capable of handling problems of potentially very large size with a minimum storage space requirement and in a reasonably fast manner.

(ii) Under the assumptions that n is large and m/n is small, which is the case in many practical situations, it is evident that the compact inverse algorithm is much preferred to the general-purpose algorithms.[2]

---

2 We did perform some numerical experiments comparing computation times required by the compact inverse and Lemke's algorithms on a number of problems similar to (2.17). We chose not to present the results here since the superiority of the former is clear from the operation counts analysis in Table 2.1, and since the reliability of such analysis had been established by the computational results shown earlier.

3. STRUCTURAL ENGINEERING APPLICATION

As mentioned in Section 1, the (parametric) LCP with a matrix of the form (1.1) appears in the nonlinear structural analysis initiated by Maier (see e.g. [12]). Maier's LCP model was later reformulated by one of the authors in terms of what he calls a n by dn LCP.

In this section, we first outline how a fundamental problem in structural engineering is formulated as the complementarity problems. After describing the principal pivoting algorithm developed in [8] to deal with the n by dn LCP, we shall present some results on comparisons of algorithms to solve the structural problem; including those of a systematic computational experiment.

3.1. The Structural Problem and Its Formulation. The formulation outlined below is essentially due to Maier. To minimize technicality, we shall treat a simplified model: we refer to Maier [12], and Kaneko [9,10] for the details and for more general approaches.

A fundamental problem in structural engineering is to determine the behavior of a structure subjected to a set of loads. It is assumed that the behavior of the entire structure can be represented by the values of stresses and strains at certain (finitely many) points in the structure.[3] We shall call them critical points and let n be the number of critical points present in the structure under consideration.

For simplicity, it is further assumed that the stress and strain at each of the critical points are one-dimensional.

The methods we shall deal with are designed for a certain particular, but practically important class of structures called (piecewise linear) elastic-plastic. In particular, reinforced concrete frames belong to this class.

Under appropriate conditions, the set of mechanical principles which governs the behavior of the elastic-plastic structure can be represented by the following (nonlinear) complementarity problem:

$$q - Ns^E + h(x) - NZNx \geq 0 \tag{3.1a}$$

3 These points may represent cross sections in a beam, or finite elements in a continuum, and so on.

(3.1b) $$x \geq 0$$

(3.1c) $$x^T(q - Ns^E + h(x) - NZNx) = 0 \quad .$$

Here, q is a nonnegative n-vector, N is a n by n diagonal matrix with plus or minus ones as diagonal elements, s^E is an n-vector such that Ns^E is positive, h is a function of the form $h(x) = (h_1(x_1),\ldots,h_n(x_n))^T$ where $h_j(x_j)$ is a concave, strictly monotone increasing real-valued function of x_j with $h_j(0) = 0$, $j = 1,\ldots,n$, and finally Z is an n by n matrix of the form $Z = -BSB^T$, where S and B are m by m and n by m, respectively.

In (3.1), all data can be determined by mechanical properties of the structure under consideration and the loads applied to it. In particular, m is the number of critical points with a certain property (called redundant) and S denotes the stiffness matrix with respect to the redundancies; thus $m < n$ and S is positive definite.

Suppose x is a solution of (3.1). From it, the vectors of stresses, s, and strains, v, are obtained by

(3.2) $$s = s^E + ZNx$$

and

(3.3) $$v = Cs + Nx \quad ,$$

where C is the n by n flexibility matrix (with respect to all critical points).

In the following, we shall attempt to explain — very briefly — some mechanical principles which give rise to (3.1).

Letting $q' = q + h(x)$ and using the expression (3.2), we may consider (3.1a) as requiring the "stress point", s, lie in the convex polyhedron

(3.4) $$E = \{s: q' - Ns \geq 0\} \quad .$$

The set E, called the elastic domain, defines the region such that the "plastic activity" takes place only if the stress point reaches the boundary of E.

The plastic activity is represented by x, the vector of plastic multipliers, and affects the structural behavior in two ways. First, it contributes to the values of stresses and strains directly through (3.2) and (3.3), respectively. The value of x

also affects the value of q, hence the shape of the elastic domain, through h. The map h is referred to as a (work-) hardening rule.

The plastic flow rules specify conditions x must satisfy; these are

$$x \geq 0 \tag{3.5}$$

$$x^T(q' - Ns) = 0 \quad . \tag{3.6}$$

The boundary hyperplanes of E represent yielding limits. Thus, the rules (3.5) and (3.6) (together with $s \in E$) state that the j^{th} plastic multiplier, x_j, can take on a positive value only if the stress point reaches the j^{th} yield limit (and yielding occurs there). The conditions (3.5)-(3.6) and that $s \in E$ give (3.1).

Finally, a remark on a parametric problem. In the representation (3.1), the effect of the applied loads is accounted for in the form of the vector s^E, which is called that of linear-elastic responses to the loads. It is often of importance to determine the complete evolution of stresses and strains during a loading process where the vector of loads of the form λf is applied at time λ for $\lambda \in [0, \bar{\lambda}]$ with $\bar{\lambda} > 0$. This can be done by solving the parametric form of (3.1), given as follows, for all λ in $[0, \bar{\lambda}]$:

$$q - \lambda Ns^E + h(x) - NZNx \geq 0 \tag{3.7a}$$

$$x \geq 0 \tag{3.7b}$$

$$x^T(q - \lambda Ns^E + h(x) - NZNx) = 0 \quad . \tag{3.7c}$$

In this section, we shall only consider, for simplicity, the nonparametric problem (3.1), except for the description of the solution procedure.

3.2. LCP Formulations. By assuming that $h_j(x_j)$ is piecewise linear, $j = 1,\ldots,n$, we can "linearize" the problem (3.1), in two alternative ways, so that pivoting methods can be applied to solve it.

First of all we note that if $h(x) = Hx$ for some (positively diagonal) matrix H, i.e. if every h_j is linear then (3.1) itself is a LCP with the matrix $H - NBSB^TN$, which can be put in the form (1.1) (a factorization of S must be obtained).

In the following, we assume that $h_j(x_j)$ has d (> 1) linear pieces as depicted in Figure 3.0 for $j = 1,\ldots,n$. Such $h_j(x_j)$ is specified by giving values of $u_j^i > 0$, $i = 1,\ldots,d$ and $r^i > 0$, $i = 1,\ldots,d-1$.

Under the above assumption on h, it can be verified (by elementary manipulations) that x solves (3.1) if and only if a dn-vector $\tilde{x}$ solves the following LCP due to Maier:

$$\tilde{q} - P^T N_s^E + (\tilde{H} - P^T N B S^T N P)\tilde{x} \geq 0 \tag{3.8a}$$

$$\tilde{x} \geq 0 \tag{3.8b}$$

$$\tilde{x}^T(\tilde{q} - P^T N_s^E + (\tilde{H} - P^T N B S^T N P)\tilde{x}) = 0 , \tag{3.8c}$$

with the relationship $x = P\tilde{x}$. In this problem, the new data, $\tilde{q} \in R^{dn}$, $P \in R^{n \times dn}$ and $\tilde{H} \in R^{dn \times dn}$, are specified as follows:

$$\left.\begin{aligned} \tilde{q}_{d(j-1)+1} &= q_j \\ \tilde{q}_{d(j-1)+k+1} &= \tilde{q}_{d(j-1)+k} + u_j^k r_j^k, \quad k = 1,\ldots,d-1 \end{aligned}\right\} \quad j = 1,\ldots,n$$

$$P = \begin{bmatrix} \overbrace{1 \ \ldots \ 1}^{d} & & 0 \\ & \ddots & \\ 0 & & \underbrace{1 \ \ldots \ 1}_{d} \end{bmatrix}$$

$$\tilde{H}_{ij} = 0, \quad i \neq j$$

$$\left.\begin{aligned} \tilde{H}_{d(j-1)+1,\, d(j-1)+1} &= u_j^1 > 0 \\ \tilde{H}_{d(j-1)+k+1,d(j-1)+k+1} &= \frac{u_j^k \, u_j^{k+1}}{u_j^k - u_j^{k-1}} > 0, \quad k = 1,\ldots,d-1 \end{aligned}\right\} \quad j = 1,\ldots,n$$

Again, after factorizing S we can put the matrix in (3.8) into the form (1.1). We would like to emphasize here that by passing from (3.1) to (3.8), the size of the problem has been increased by a factor of d.

The linearization scheme employed by one of the authors [10] is based on a representation of the piecewise linear function h given by

$$h(x) = \sum_{i=1}^{d} U^i x^i , \tag{3.9}$$

with conditions

$$0 \leq x^i \leq r^i, \quad i = 1,\ldots,d-1, \quad x^d \geq 0 \tag{3.10}$$

$$(r^i - x^i)^T x^{i+1} = 0, \qquad i = 1,\ldots,d-1 \tag{3.11}$$

and

$$x = \sum_{i=1}^{d} x^i ,$$

where $x^i \in R^n$, $r^i = (r^i_1,\ldots,r^i_n)^T$, and $U^i = \text{diag}\{u^i_1,\ldots,u^i_n\}$, $i = 1,\ldots,d$.

Substituting h in (3.1) by (3.9) along with the side conditions yields the n by dn LCP[4]:

$$q - Ns^E + \sum_{i=1}^{d} (U^i - NZN)x^i \geq 0 \tag{3.12a}$$

$$0 \leq x^i \leq r^i, \quad i = 1,\ldots,d-1, \quad x^d \geq 0 \tag{3.12b}$$

$$x^1(q - Ns^E + \sum_{i=1}^{d} (U^i - NZN)x^i) = 0 \tag{3.12c}$$

$$(r^i - x^i)^T x^{i+1} = 0, \quad i = 1,\ldots,d-1 . \tag{3.12d}$$

Since $-Z$ is positive semi-definite and U^i is diagonal with positive diagonal elements[5], $i = 1,\ldots,d$, it follows that the n by dn matrix

$$(U^1 - NZN, \ldots, U^d - NZN)$$

has the $\mathcal{P}$-property (see Kaneko [8]) and thus (3.12) has a unique solution, which can be computed by the algorithm developed in [8] (described in the next subsection).

The obvious advantage of the n by dn formulation (3.12) is that it involves only n linear constraints, except for the simple upper bounds on some of the variables which can be treated as in the standard upper bounding technique in linear programming.

3.3. The Condensed Graves' Algorithm. In the following, we shall describe the principal pivoting algorithm developed by Kaneko [8] to solve an n by dn LCP with a matrix

---

4 Note that under the conditions (3.10)-(3.11), the complementarity (3.1c) is equivalent to (3.12c).

5 The fact that $u^1_j > \ldots > u^d_j > 0$, $j = 1,\ldots,n$, follows from the assumed monotonicity and concavity of $h_j(x_j)$.

having the $\mathscr{P}$-property. The algorithm was called the condensed Graves' algorithm for the reason explained in [8]. The algorithm given below is essentially the same as in [8] except that in [8] d is assumed to be two while here it can be any positive integer.

The algorithm is applied to a parametric form of an n by dn LCP:

$$w = q + \lambda p + \sum_{i=1}^{d} M^i x^i \tag{3.13a}$$

$$w \geq 0, \quad 0 \leq x^i \leq r^i, \quad i = 1,\ldots,d-1, \quad x^d \geq 0 \tag{3.13b}$$

$$w^T x^1 = 0, \quad (r^i - x^i)^T x^{i+1} = 0, \quad i = 1,\ldots,d-1 \ , \tag{3.13c}$$

where $\lambda \in [0, \bar{\lambda}]$, $\bar{\lambda} > 0$.

In a general iteration step, there exist index sets $\gamma^0,\ldots,\gamma^d$ such that

$$\begin{cases} w_{\gamma^0}: \text{ basic} \\ w_{\gamma^j} = 0: \text{ nonbasic}, \quad j = 1,\ldots,d \end{cases}$$

and for $i = 1,\ldots,d$

$$\begin{cases} x^i_{\gamma^j} = 0: \text{ nonbasic}, \quad j = 0,\ldots,i-1 \\ x^i_{\gamma^i}: \text{ basic} \\ x^i_{\gamma^j} = r^i_{\gamma^j}: \text{ nonbasic}, \quad j = i+1,\ldots,d \ . \end{cases}$$

We shall denote by q,p,M, the current values of the data. Also we denote by the n-vector defined by

$$y = \sum_{i=1}^{d-1} \sum_{j \in \gamma^i} r^i_j M^i_{.j} \ .$$

Algorithm II. Condensed Graves' algorithm.

Step 1: If $p_{\gamma^0} \geq 0$, $p_{\gamma^d} \geq 0$ and $p_{\gamma^j} = 0$, $j = 1,\ldots,d-1$, then stop: the curre basis gives the solution at $\lambda = \bar{\lambda}$.

Step 2: Determine the next critical index s by

$$s \in \arg\min \left\{ \frac{-(q_j+y_j)}{p_j} : \; p_j < 0, \quad j = 1,\ldots,n \; ; \right.$$

$$\left. \frac{r_j - (q_j+y_j)}{p_j} : \; p_j > 0, \quad j \in \bigcup_{i=1}^{d-1} \gamma^i \right\} .$$

The next critical value, λ', is given by

$$\lambda' = \begin{cases} \dfrac{-(q_s+y_s)}{p_s} & \text{if} \quad p_s < 0 \\ \dfrac{r_s-(q_s+y_s)}{p_s} & \text{if} \quad p_s > 0 \end{cases} .$$

Step 3: If $\lambda' \geq \bar{\lambda}$, then stop: the current basis gives the solution at $\lambda = \bar{\lambda}$.

Step 4: Let s be the critical index determined in Step 2. The next pivot element is $M^{i^*}_{ss}$, where i^* is given by the following:

(i) $p_s < 0$, $s \in \gamma^i$, $i \in \{0,1\}$, then $i^* = 1$

(ii) $p_s < 0$, $s \in \gamma^i$, $i \in \{2,\ldots,d\}$, then $i^* = i-1$

(iii) $p_s > 0$, $s \in \gamma^i$, $i \in \{1,\ldots,d-1\}$, then $i^* = i+1$.

Determine the new "eta-vector" by computing the current value of $M^{i^*}_{.s}$ ("forward transformation"). Update q, p and y by standard pivot formulas. Return to Step 1.

3.4. Comparison of Algorithms. Suppose that we are to solve the problem of determining the behavior of a structure with n critical points of which m are "redundant" and $h_j(x_j)$ having $d > 1$ linear pieces, $j = 1,\ldots,n$. Our objective is to compare the computer storage space and the number of operations required to solve the problem by (i) Lemke's algorithm (on (3.8)), (ii) the compact inverse algorithm (on (3.8)) or (iii) the condensed Graves' algorithm (on (3.12)).

The results of the comparison are summarized in Table 4.1. To make the comparison, it is assumed that specifications of the first two algorithms are as stated in Section 2, and those of the third one are the same as Lemkes'.

It is clear that for $d > 1$ the condensed Graves' algorithm is always superior to Lemkes' with respect to both criteria, provided that T' and T are close[6]. Thus,

6 Refer to footnote 1.

	Lemke	compact inverse	condensed Graves
storage	$d^2n^2 + T'dn + dn$	$dmn + 3dn + \frac{m(m+1)}{2}$	$n^2 + Tn + 2dn$
operation counts	$dn\frac{T'(T'-1)}{2} + 3T'dn$	$T\left[dn(m+3)+2(m+1)^2\right]$	$n\frac{T(T-1)}{2} + 4Tn$

Table 3.1. Storage and operation counts for three algorithms

Lemke's algorithm will be omitted from further considerations.

The relative advantage of the two special-purpose algorithms depend on the values of n,m,d and T. We shall discuss, below, likely values of n,m,d and T in practical problems, but in general the following may be asserted: The condensed Graves' algorithm is more advantageous when m/n is large and T is small, and the compact inverse algorithm is more advantageous when m/n is small and T is large.

In this particular application, the matrices involved (in (3.8) and (3.12)) tend to be highly dense.

The value of n could be quite large if for instance the underlying structure is continuum. In practical reinforced concrete frame problems, small to medium size n is expected.

The value of m varies from one problem to another, but the ratio m/n is expected to be in the neighborhood of one half in most cases.

The value of d depends on how accurately the piecewise linear approximation of h must be done. In reinforced concrete frame problems, for example, d = 2 or 3 seems adequate.

Finally, in this application the number of pivots, T, is expected to be of a moderate magnitude, even when the parametric problem (3.7) must be solved, the range of λ is relatively small. The actual value of T could be highly variable but in general it is expected to be large when dn is large (note that in both (3.8) and (3.12) there are dn variables).

3.5. Computational Results. Based on the comparison in terms of the operation counts, it is expected that the condensed Graves' algorithm solves the problem more quickly than the compact inverse algorithm, when m/n is large and the relative advantage is reversed when m/n becomes small (such a value of m/n will be called the reversal value).

In order to (i) verify this tendency and (ii) determine the reversal value of m/n, we have performed a computational experiment comparing the computational times required by the two algorithms to solve some problems of various sizes. The results are reported below.

We first constructed and solved the n by dn LCP (3.12) by the condensed Graves' algorithm for the sets of data with various values of d, n and m (see below). The data were randomly generated in the following form with the specified properties:

$$q \geq 0, \quad p = -Ns^E < 0,$$

$$M^i = U^i + GG^T \text{ with } n \text{ by } m \; G \text{ and}$$

$$U^i = \text{diag}\{u_1^i, \ldots, u_n^i\}, \quad i = 1, \ldots, d, \text{ where}$$

$$u_j^1 > \ldots > u_j^d > 0, \quad j = 1, \ldots, n$$

$$r^i > 0, \quad i = 1, \ldots, d-1 .$$

We then converted the data to obtain the corresponding LCP (3.8), and solved it by the compact inverse algorithm.

A total of 147 problems were solved by each of the two algorithms with the following values of d, n and m; problems with the same values of d and n are considered to be in one group.

Group 1: $d = 2$, $n = 40$, $m = 2,4,28,30$ and $m = 8 + 2t$, $t = 0,\ldots,6$

Group 2: $d = 2$, $n = 50$, $m = 2,4,38,40$ and $m = 8 + 2t$, $t = 0,\ldots,10$

Group 3: $d = 2$, $n = 60$, $m = 2,4,48,50$ and $m = 8 + 2t$, $t = 0,\ldots,13$

Group 4: $d = 2$, $n = 70$, $m = 2,4,48,50$ and $m = 10 + 2t$, $t = 0,\ldots,15$

Group 5: $d = 3$, $n = 40$, $m = 2,4,28,30$ and $m = 8 + 2t$, $t = 0,\ldots,6$

Group 6: $d = 3$, $n = 50$, $m = 2,4,38,40$ and $m = 9 + 3t$, $t = 0,\ldots,7$

Group 7: $d = 3$, $n = 60$, $m = 2,4,48,50$ and $m = 9 + 3t$, $t = 0,\ldots,7$

Group 8: $d = 3$, $n = 70$, $m = 2,4,48,50$ and $m = 9 + 3t$, $t = 0,\ldots,9$.

We wrote a Fortran program for the condensed Graves' algorithm which was based on the specifications stated in Section 4.3. The program for the compact inverse algorithm used was essentially the same as that mentioned in Section 3, except for a certain modification which takes into account the special forms of the matrices P and N.

Results of the experiment are, in general, closely in accordance with the analysis based on the operation counts. Some of them are shown and explained in the following.

Figures plotting the computation times (exclusive of input/output) per iteration in both algorithms are drawn for problems in each of the eight groups. Figures 3.1-3.3 are those for Groups 1, 4 and 7; others are omitted since they appear quite similar to those shown here. The above-mentioned tendency of the relative advantage of the algorithms is clearly demonstrated in all the figures.

These figures also confirms a result of the analysis in Table 3.1 that the time per iteration in the compact inverse algorithm increases quadratically in m. Figure 3.4 shows that the time per iteration in the condensed Graves' algorithm increases linearly in T, confirming the corresponding analytical result in Table 3.1.

In Figures 3.1-3.3 the time per iteration in the condensed Graves' algorithm appears to be decreasing in m and thus in conflict with the analytical result (which is independent of m). This was, however, a natural consequence of the fact that T, the number of iterations needed to solve the problem, tended to be smaller as m became larger;[7] recall that the time per iteration increases linearly in T.

Approximate values of reversal were determined in the following fashion. In Figures 3.1-3.3 (and in similar ones not shown here) we drew horizontal lines to indicate the time per iteration in the condensed Graves' algorithm corresponding to a problem which is solved in $\bar{T}$ iterations, where $\bar{T}$ is the average of the numbers of iterations necessary to solve the problems in Group 1, 4 and 7, respectively. We need to do so since the time per iteration in the condensed Graves' algorithm depends linearly in T (as shown in Figure 3.4) and the number of iterations needed to solve each of the problems in each group varied.

---

7 This tendency may be attributed to the manner in which the data were constructed; we do not elaborate on this point here, however.

In each of Figures 3.1-3.3 let m^* be the value of m corresponding to the "cross-over" point, i.e. the point at which the horizontal line meets the quadratic curve corresponding to the time for compact inverse algorithm. For the reasons mentioned above, it is appropriate to consider m^*/n as an approximate value of reversal. Table 3.2 lists approximate values of reversal obtained in this fashion along with other information for problems in each group.

Group	1	2	3	4	5	6	7	8
reversal value $\frac{m^*}{n}$	.3	.3	.3	.3	.4	.5	.4	.4
average # of iterations	38	51	70	82	71	100	83	83
d	2	2	2	2	3	3	3	3
n	40	50	60	70	40	50	60	70

Table 3.2. Approximate reversal values.

We parenthetically mention that the reversal value tends to be large when the average number of iterations is large; this is consistent to the analytical result in Table 3.1.

For further computational results, we refer to a report [10] by one of the authors establishing the predicted superiority of the condensed Graves' algorithm over Lemkes' with respect to the computation time.

3.6. Conclusions. In the structural engineering application, we can draw the following conclusions on the suitability of the solution methods.

(i) The condensed Graves' algorithm is always better than Lemke's algorithm, both in terms of the computer storage and computation time required to solve the problem.

(ii) In most problems (where m is about $n/2$), the condensed Graves' algorithm solves the problem more quickly requiring more storage space than the compact inverse algorithm.

(iii) The compact inverse algorithm is better than the condensed Graves' algorithm in terms of both storage and computer time if m/n is small and dn is large.

4. GRADUATION APPLICATION

4.1. Problem Description. In this section, we discuss the third and final application of the class of LCP's and PLCP's with matrices of the form (1.1). This application arises in the area of graduation or curve fitting and smoothing. Before describing how the complementarity problems emerge we explain the meaning and process of graduation in the context of actuarial sciences. The following discussion is based largely on the excellent treatise of the subject, a monograph by M. D. Miller [18].

The actuary is concerned with the contingencies of death, disability, retirement, sickness, withdrawal, marriage, etc. He must know the probabilities of such events in order to predict their future occurrence and in order to be able to calculate premiums, reserves, annuities and so forth, for insurance and other financial institutions. Tables setting forth such probabilities must be constructed and for that purpose, observations are made of the happening of such events. Graduation is one of the steps in the construction of these tables.

Typically, a series of observed probabilities is found to contain irregularities. It is with irregular series of observed values of continuous varying quantities such as mortality rates, that the problem of graduation deals. Formally, graduation may be defined as the process of securing from an irregular series of observed values of a continuous variable a smooth, regular series of values consistent in a general way with the observed series of values. This smooth series of graduated values is then taken as a representation of the underlying law which gave rise to the series of observed values.

There are several theoretical reasons justifying the postulate of a smooth regular underlying series of true values in the process of graduation. One of these reasons can be dealt with using the mathematical theory of probability, see [18].

Among those methods by which the process of graduation may be accomplished is the difference-equation method. This method is founded on the formulation of an analytic expression measuring the combination of two essential characteristics of graduation: (i) smoothness and (ii) fitness, or consistency with the observed data. On the one hand, the graduated series should be smooth compared with the ungraduated series; on the other hand it should be consistent with the indications of the ungraduated series. Basically,

these two qualities of smoothness and fitness are inconsistent in the sense that smoothness may not be improved without some sacrifice of fitness, and vice versa. This situation is similar to Markowitz' mean-variance model in portfolio analysis.

Numerically, smoothness and fitness are measured, respectively, by

$$F = \sum_{x=1}^{n} w_x (u_x - u''_x)^2 \tag{4.1}$$

and

$$S = \sum_{x=1}^{n-m} (\Delta^m u_x)^2 \tag{4.2}$$

where

u''_x = ungraduated values for integral values $x = 1$ to $x = n$

u_x = corresponding graduated value

w_x = weight assigned to the value u''_x

$\Delta^m u_x$ = m-th forward difference of the graduated series

$$= \sum_{r=0}^{m} (-1)^{m-r} \binom{m}{r} u_{x+r}$$

and

m = integer indicating the desired degree of smoothness; typical values are 2, 3, 4 and 5 .

If the ungraduated values u''_x represent the mortality rates, then the weights w_x are normally assigned the corresponding numbers exposed to the risk of death.

For a specified value of m, the difference-equation graduation formula, also referred to as Whittaker-Henderson formula, is derived from the minimization of the expression $F + kS$ where k is the constant indicating the relative emphasis we are placing on fitness and smoothness. Before the advent of electronic computers, it was generally felt that this graduation method, while producing excellent results, involved a working process so long and difficult as to be impractical except for making graduations of considerable importance. Nevertheless, as it has been pointed out in [7], this method is really very simple, especially when vector-matrix notation is employed. Indeed, letting u" and u

be the vectors of ungraduated and graduated values respectively, Σ be the diagonal matrix of weights, and G be the coefficient matrix of the m-th order difference operator Δ^m, i.e.

(4.3) $$G = \begin{bmatrix}
(-1)^m & & 0 & & \\
(-1)^{m-1}\binom{m}{1} & (-1)^m & & 0 & \\
(-1)^{m-2}\binom{m}{2} & (-1)^{m-1}\binom{m}{1} & & & \\
\vdots & \vdots & \ddots & & 0 \\
1 & & & (-1)^m & 0 \\
0 & 1 & & (-1)^{m-1}\binom{m}{1} & (-1)^m \\
0 & 0 & & \vdots & \\
\vdots & \vdots & & & \ddots \\
0 & 0 & & 1 & \\
\vdots & \vdots & & \vdots &
\end{bmatrix}$$

we have

(4.4) $$F + kS = u^T(\Sigma + kGG^T)u - 2u^T \Sigma u'' + u''^T \Sigma u'' .$$

Differentiation with respect to u gives

(4.5) $$(\Sigma + kGG^T)u = \Sigma u''$$

from which the graduated series (u_x) can be obtained readily. Observe that G is of order n by $(n-m)$ and the nonzero elements of each of its column are defined by the $(m + 1)$ binomial coefficients $(-1)^m$, $(-1)^{m-1}\binom{m}{1}$, ..., 1. Moreover, each row of G contains at most $(m + 1)$ non-zero entries.

Summing up the above discussion, we see that the difference-equation approach for graduation involves nothing but the solution of a system of linear equations (4.5) derived from an unconstrained minimization problem.

Despite the fact that this particular approach for graduation has been widely accepted and used by researchers and practitioners in actuarial sciences, we feel that it is imperfect and could be improved. In particular, we propose the following constrained difference-equation approach: Let C denote the set of a priori information that we have about the series of ungraduated values u''_x, i.e. C is a given subset of R^n containing the vector u''. The new approach requires that the graduated series u be the solution to the following constrained minimization problem

$$\text{minimize } F + kS \text{ subject to } u \in C . \tag{4.6}$$

The reason for considering such a constrained problem is clear. This is because we feel that the graduated series should, in addition to exhibiting some consistency and the same tendency as the ungraduated series, possess the same characteristics as the latter series. For instance, if we are graduating mortality rates, we certainly hope that the graduated values will represent probabilities. By taking C to be the unit cube in (4.6), we are guaranteed that the graduated values will indeed be nonnegative and not greater than one. It should be pointed out that requiring the graduated values to be probabilities by no means implies that the graduated series is a probability density function. Therefore, no restriction on the sum of graduated values is necessary.

In the important case where C is a rectangle, we can, by an easy change of variables, cast problem (4.6) in the form below:

$$\text{minimize } \frac{1}{2} u^T(\Sigma + kGG^T)u + q^T u \text{ subject to } \quad 0 \leq u \leq a \quad . \tag{4.7}$$

To the authors' knowledge, problem (4.7) has not explicitly appeared in the literature. Nevertheless, several of its special cases have been studied under various contexts. Of course, without the constraints, problem (4.7) reduces to the classical difference-equation approach. Incidentally, the unconstrained problem has also been studied in the context of spline functions, see [13], [20], [21], [23]. Mangasarian and Schumaker [14] have provided necessary and sufficient optimality conditions for the case $\Sigma = 0$ in terms of difference operators. The continuous version of this special case has appeared in [15] under the context of control theory and in [3] under the context of an engineering application.

The Karush-Kuhn-Tucker optimality conditions for (4.7) are given by:

$$w = q + (\Sigma + kGG^T)u + v \geq 0, \quad u \geq 0 \tag{4.8a}$$

$$z = a - u \geq 0, \quad v \geq 0 \tag{4.8b}$$

$$w^T u = z^T v = 0 \ . \tag{4.8c}$$

If the upper bounds a_i are all infinity, then problem (4.8) reduces to a linear complementarity problem (q,M) where M is of the form (1.1). Therefore the compact inverse algorithm is directly applicable in this case. The general case where not all the a_i's are infinity can be treated by an extension of the algorithm. See [19].

In what follows, we point out some characteristics of the matrix $\Sigma + kGG^T$ appearing in this graduation application which can be used advantageously in the application of Algorithm I (and its extension). Since m is typically very small, the n by (n-m) matrix is almost square. According to the comparison table in Section 2.3, this latter fact would seem to imply that the application of the compact inverse algorithm would be undesirable. Nevertheless, (4.3) shows that G is very sparse and specially structured. Indeed, it is easy to see that for every β in $\{1,\ldots,n\}$, the matrix $A(\beta)$ defined in (2.9) is a band matrix with band width $2m + 1$. Therefore the Cholesky factor L has at most m nonzero off-diagonal elements in each row. The sparsity of L reduces the amount of operations in Subroutines I and II from $2(n-m)^2 + 4(n-m)$ to $n(4m+6) - 7m(m+1)$. Moreover, the fact that the nonzero elements in each row of G are coefficients of the m-th order difference operator Δ^m makes it clear that the vector $\xi^{\beta,k}$ defined by (2.22) can be calculated without any multiplication. Finally, because of its special structure, no storage space is required for the matrix G in the implementation of the algorithm. Its elements can be generated very easily if needed.

The matrix $M = \Sigma + kGG^T$ is sparse if G is (although M is less sparse than G), and this could certainly be taken into account to increase the efficiency of general-purpose algorithms such as Lemke's. However, the fact that the compact inverse algorithm could be implemented so that it takes full advantage of the special structures of G indicates the possibility that the special-purpose algorithm may turn out to be more efficient than general-purpose algorithms. In order to draw empirical conclusions, we have performed several computational experiments. The results are reported in the next subsection.

4.2. Computation Results. In order to compare the efficiencies of the compact inverse and Lemke's algorithms in solving (P) LCP's with matrix of the form (1.1), where G is given explicitly by (4.3), experiments are designed to solve the following variant of problem (4.7):

$$\text{(4.9)} \qquad \text{minimize } \frac{1}{2} u^T(\Sigma + kGG^T)u + q^T u \quad \text{subject to} \quad u \geq 0 .$$

We do not consider upper bounds in the experiments because the code NULEMKE that we use for Lemke's algorithm handles upper bounds as separate constraints, whereas the compact inverse algorithm treats them implicitly. It is certainly disadvantageous to Lemke's algorithm if the upper bounds are included. It should be mentioned that NULEMKE does take advantage of sparsity of the data.

A set of three experiments corresponding to $m = 2,3$ and 4 are performed. For each value of m, five values 50, 60, 70, 80 and 90 of n are tested.[8] In the case $m = 3$, it is found that, if not for the reinversion in Lemke's algorithm, which occurs when $n = 90$, the two algorithms produce results very close to each other. The results for $m = 2$ and 4 are summarized in Tables 4.1 and 4.2.

n	density of M (%)	# of iterations[1]	time/iteration[2] (10^{-3} sec.) Lemke	compact inverse
50	9.8	48	11.6	14.0
60	8.2	58	15.1	16.7
70	7.0	67	16.7	19.3
80	6.2	76	18.9	21.9
90	5.5	88	29.3[3]	24.7

Table 4.1 $m = 2$

8 These values of m and n are typical in actuarial graduation.

n	density of M (%)	# of iterations[1]	time/iteration[2] (10^{-3} sec.) Lemke	compact inverse
50	17.2	58	18.1	18.6
60	14.4	70	26.0	23.0
70	12.5	84	27.8	25.4
80	10.9	109	36.7[3]	29.6
90	9.7	114	38.2[3]	33.3

Table 4.2 m = 4

Notes: 1.We chose the covering vector in Lemke's algorithm so that the numbers of iterations in both algorithms are the same.

2. The time is exclusive of input/output.

3. The sudden increase is due to the occurrence of reinversion of the basis.

4.3. Conclusions. Based on the above computational results, we may draw the following three conclusions:

(i) The compact inverse algorithm requires much less storage space than Lemke's algorithm.

(ii) Although the ratio $(n-m)/n$ is close to one in these graduation problems, the compact inverse algorithm is performing compatibly with Lemke's algorithm, in terms of computation time.

(iii) The results indicate that as m gets larger, or equivalently, as G becomes "thinner", the efficiency of the compact inverse algorithm becomes higher, a fact consistent with the analysis of the algorithm made in Section 2.3.

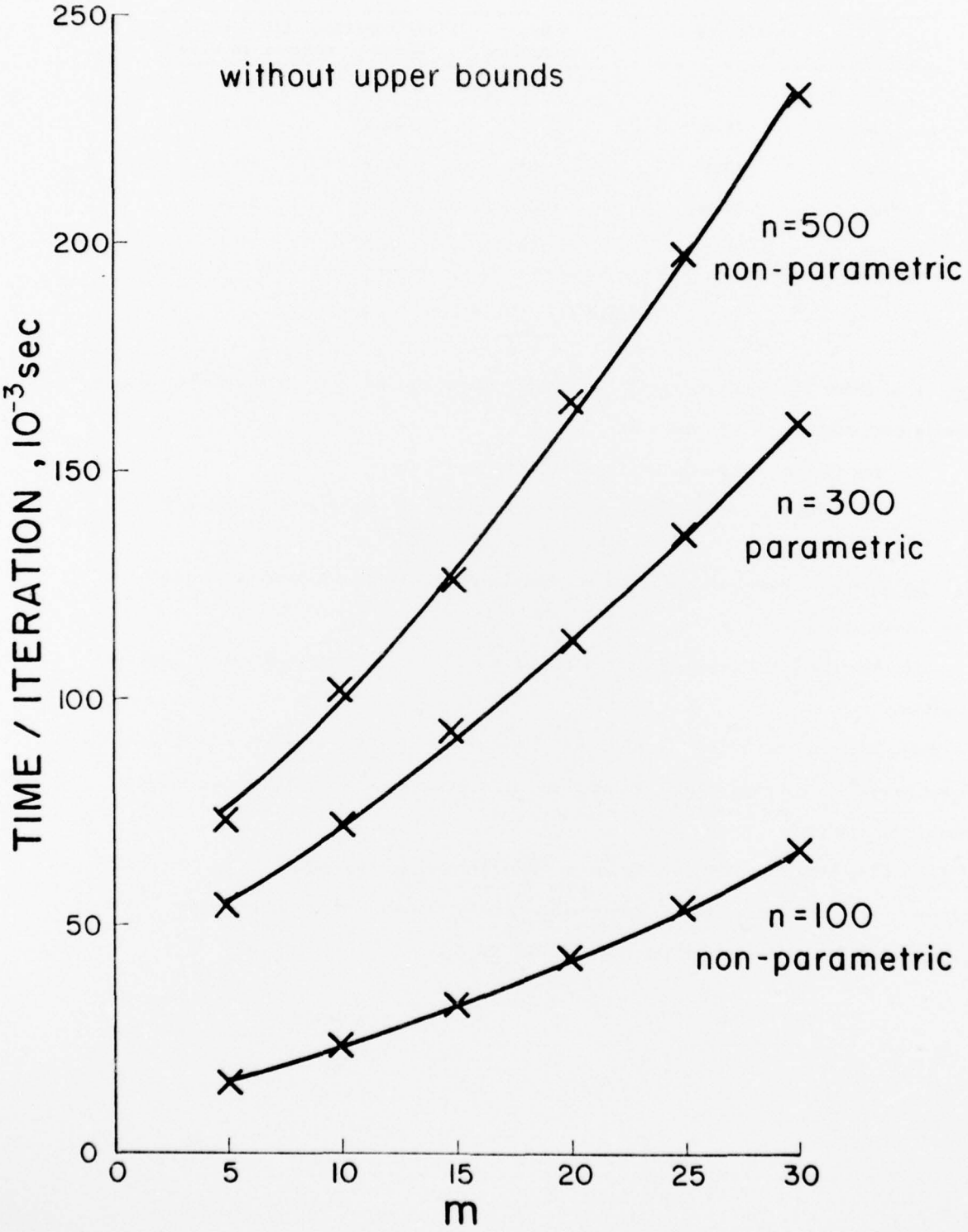

without upper bounds
n=500
non-parametric
n = 300
parametric
n=100
non-parametric
TIME / ITERATION , 10^{-3}sec
250
200
150
100
50
0
0
5
10
15
20
25
30
m

Figure 2.1

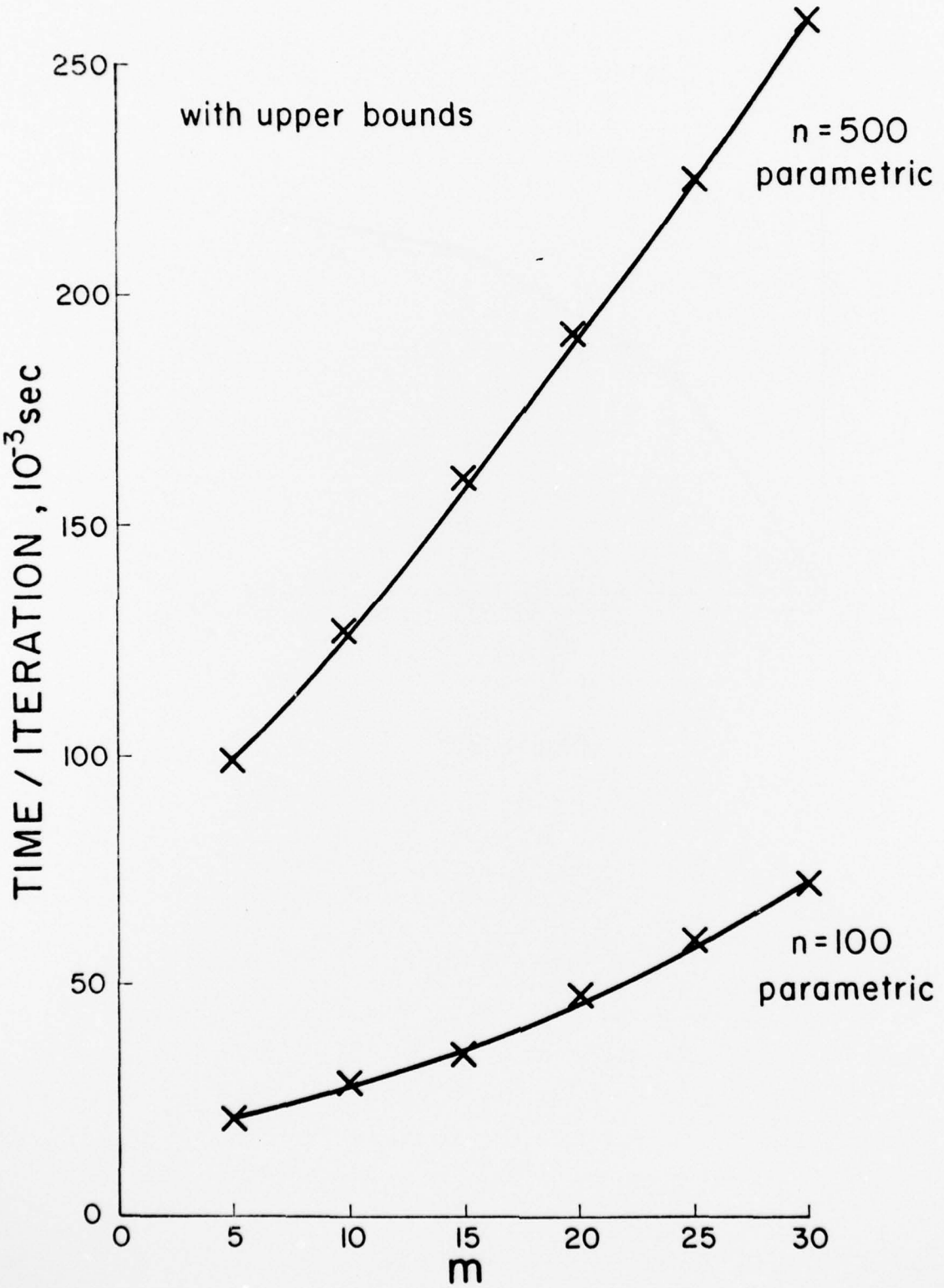

with upper bounds
n = 500
parametric
n = 100
parametric
TIME / ITERATION , 10^{-3} sec
250
200
150
100
50
0
0
5
10
15
20
25
30
m

Figure 2.2

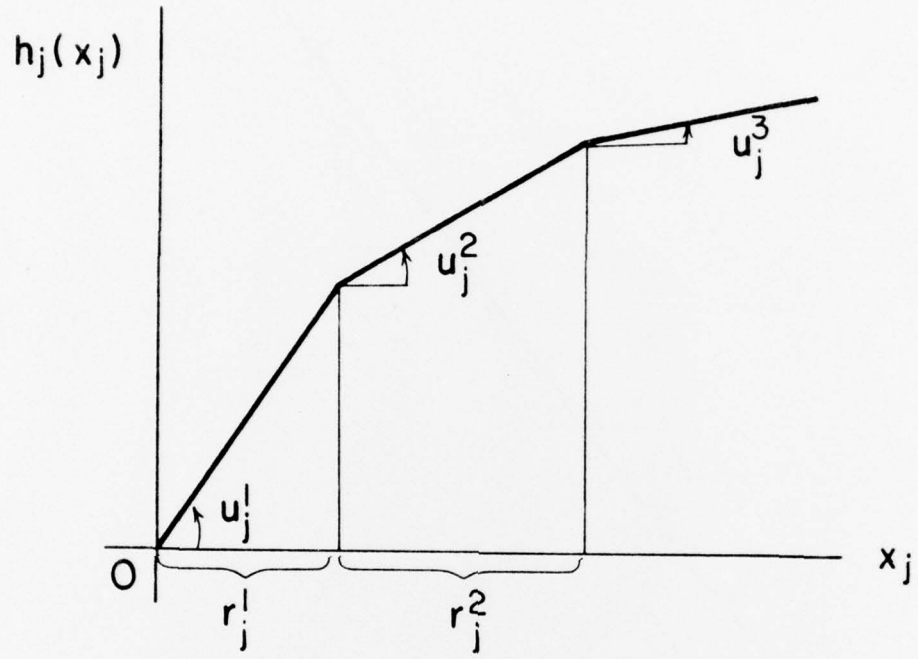


Figure 3.0

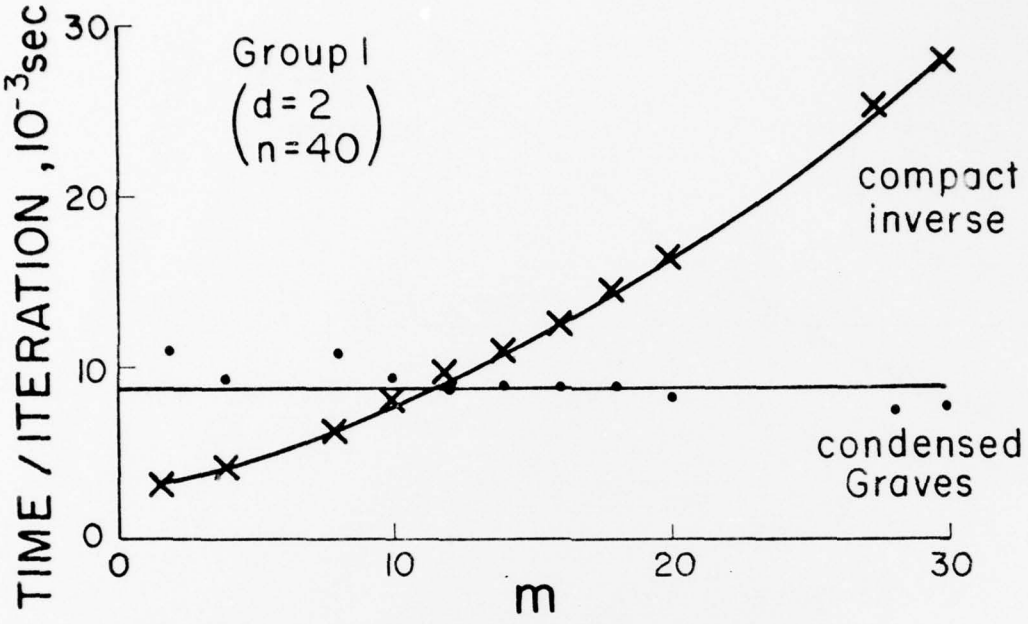


Figure 3.1

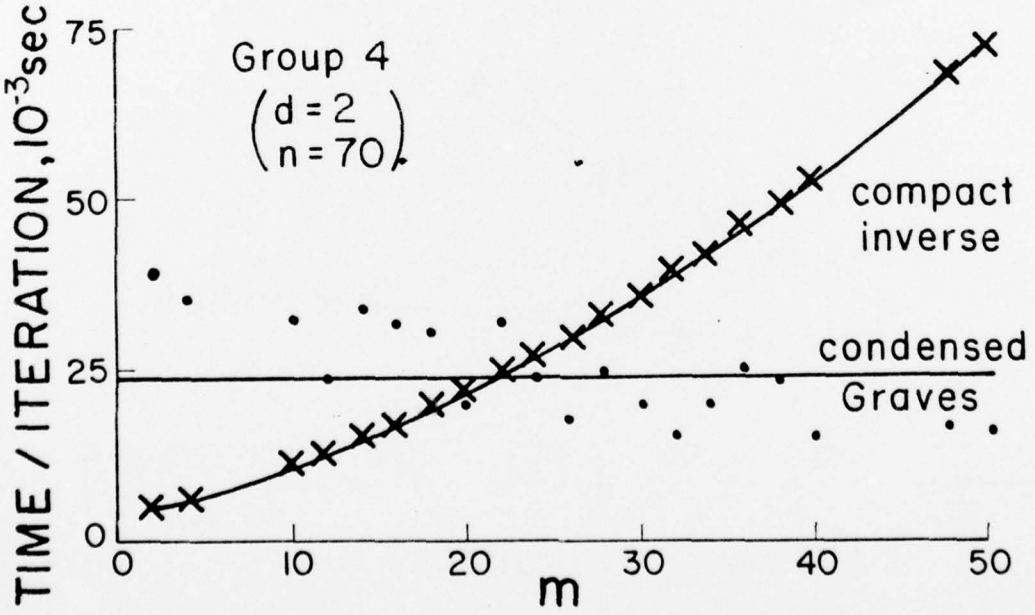


Figure 3.2

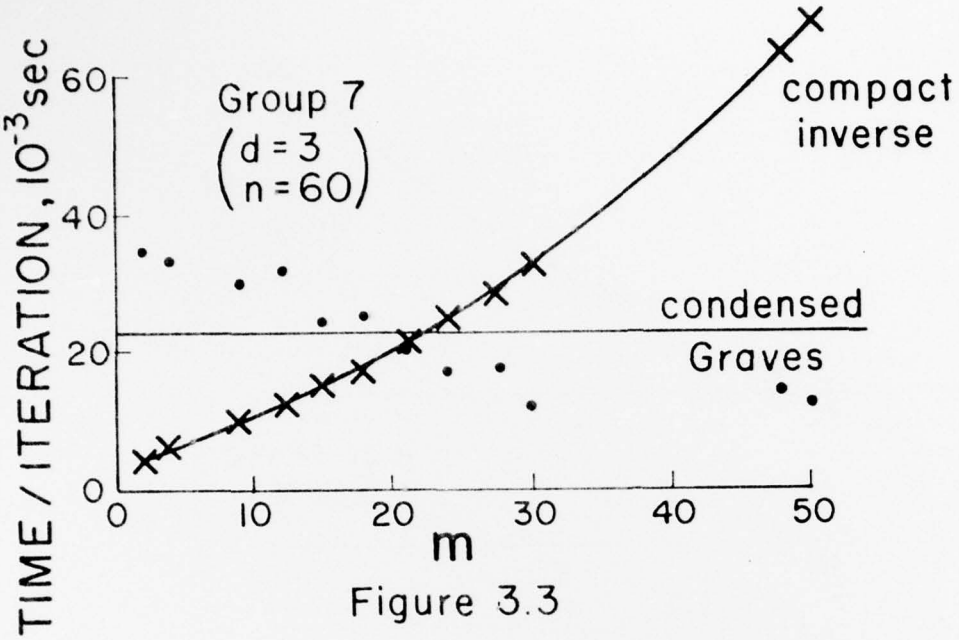


Figure 3.3

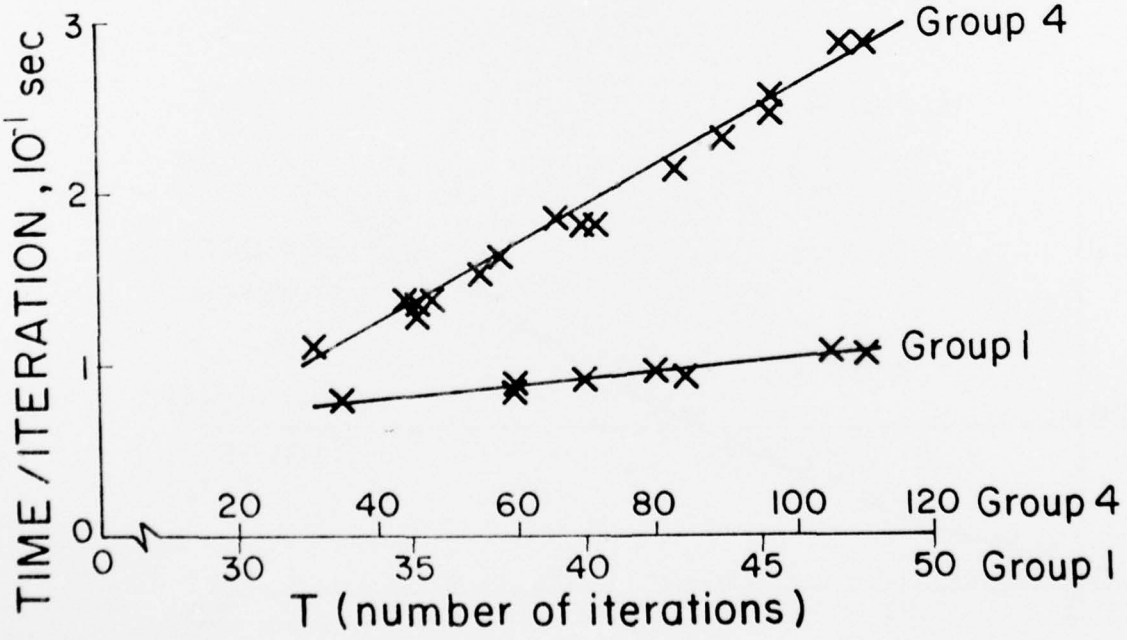


Figure 3.4

REFERENCES

[1] R. W. Cottle, "Monotone solutions of the parametric linear complementarity problem", Mathematical Programming 3(1972), 210-224.

[2] R. W. Cottle and G. B. Dantzig, "Complementary pivot theory and mathematical programming", Linear Algebra and its Applications 1(1968), 103-125.

[3] U. Eckhardt, "Quadratic programming by successive overrelaxation", Technical report Jül-1064-MA, Kernforschungsanlage, Jülich 1974.

[4] J. J. H. Forest and J. A. Tomlin, "Updated triangular factors of the basis to maintain sparsity in the product form simplex method", Mathematical Programming 2(1972), 263-278.

[5] P. E. Gill, G. H. Golub, W. Murray and M. A. Saunders, "Methods for modifying matrix factorizations", Mathematics of Computation 28(1974), 505-535.

[6] R. L. Graves, "A principal pivoting simplex algorithm for linear and quadratic programming", Operations Research 15(1967), 482-494.

[7] T. N. E. Greville, "Graduation", Part 5 Study Notes, The Actuarial Society of America, 1974 Edition.

[8] I. Kaneko, "A linear complementarity problem with an n by 2n "P"-matrix", Technical report WP76-4, Department of Industrial Engineering, University of Wisconsin-Madison, February 1977. To appear in Mathematical Programming.

[9] I. Kaneko, "Complete solutions of a class of elastic-plastic structures", Technical report WP77-17, Department of Industrial Engineering, University of Wisconsin-Madison, June 1977.

[10] I. Kaneko, "Piecewise linear elastic-plastic analysis", Technical report WP77-18, Department of Industrial Engineering, University of Wisconsin-Madison, April 1977.

[11] C. E. Lemke, "Bimatrix equilibrium points and mathematical programming", Management Science 11(1965), 681-689.

[12] G. Maier, "A matrix structural theory of piecewise-linear elastoplasticity with interacting yield-planes", Meccanica 5(1970), 54-66.

[13] M. A. Malcolm, "On the computation of nonlinear spline functions", SIAM Journal on Numerical Analysis 14(1977), 254-282.

[14] O. L. Mangasarian and L. L. Schumaker, "Discrete splines via mathematical programming", MRC Technical report #1028, Mathematics Research Center, University of Wisconsin-Madison, January 1970.

[15] O. L. Mangasarian and L. L. Schumaker, "Splines via optimal control", in I. J. Schoenberg (ed.) "Approximations with special emphasis on spline functions", Academic Press, New York 1969, 119-156.

[16] H. M. Markowitz, "Portfolio selection", The Journal of Finance 12(1952), 77-91.

[17] H. M. Markowitz, "The optimization of a quadratic function subject to linear constraints", Naval Research Logistics Quarterly 3(1956), 111-133.

[18] M. D. Miller, "Elements of graduation", The Actuarial Society of America, 1949.

[19] J. S. Pang, "Some new and efficient algorithms for portfolio analysis", MRC Technical report #1738, Mathematics Research Center, University of Wisconsin-Madison, March 1977.

[20] C. H. Reinsch, "Smoothing by spline functions", Numerische Mathematik 10(1967), 177-183.

[21] C. H. Reinsch, "Smoothing by spline functions. II", Numerische Mathematik 16(1971), 451-454.

[22] H. Samelson, R. M. Thrall and O. Wesler, "A partitioning theorem for Euclidean n-space", Proceedings American Mathematical Society 9(1958), 805-807.

[23] I. J. Schoenberg, "Spline functions and the problem of graduation", Proceedings National Academy of Sciences (U.S.A.) 52(1964), 947-950.

[24] W. F. Sharpe, "A simplified model for portfolio analysis", Management Science 9(1963), 277-293.

[25] W. F. Sharpe, "Portfolio theory and capital markets", New York: McGraw-Hill, 1970.

[26] B. Von Hohenbalken, "A finite algorithm to maximize certain pseudo-concave functions on polytopes", Mathematical Programming 9(1975), 189-206.

[27] P. Wolfe, "The simplex method for quadratic programming", Econometrica 27(1959), 382-398.

(9 Technical summary rept.)

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: (6) 1785 (14) MRC-TSR-1785 ; 2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): ON THE SOLUTION OF SOME (PARAMETRIC) LINEAR COMPLEMENTARITY PROBLEMS WITH APPLICATIONS TO PORTFOLIO ANALYSIS, STRUCTURAL ENGINEERING AND GRADUATION.	5. TYPE OF REPORT & PERIOD COVERED: Summary Report - no specific reporting period 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): (10) Jong-Shi Pang, Ikuyo Kaneko and Wayne P. Hallman	8. CONTRACT OR GRANT NUMBER(s): MCS75-17385, (15) DAAG29-75-C-0024, NSF-MCS-75-17385
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Mathematics Research Center, University of Wisconsin, 610 Walnut Street, Madison, Wisconsin 53706	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: Work Unit Number 5 - Mathematical Programming and Operations Research
11. CONTROLLING OFFICE NAME AND ADDRESS: See Item 18 below	12. REPORT DATE: (11) August 1977 13. NUMBER OF PAGES: 40 (12) 44p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

U.S. Army Research Office
P.O. Box 12211
Research Triangle Park
North Carolina 27709

and

National Science Foundation
Washington, D.C. 20550

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Applications, (parametric) linear complementarity, quadratic programming, portfolio analysis, index models, structural engineering, graduation, difference-equation method, general-purpose algorithms, special-purpose algorithms, comparison, computational results

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

In this paper we discuss three important applications of a class of (parametric) linear complementarity problems arising independently from such diverse areas as portfolio analysis, structural engineering and graduation. After explaining how the complementarity problems emerge in these applications, we perform some analytical comparisons (based on operation counts and storage requirements) of several existing algorithms for solving this class of complementarity problems. We shall also present computational results to support the analytical comparisons, finally, we derive some conclusions about the general performance of these algorithms.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

221200